NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

FORM ASSEMBLY FOR THE ARMED SERVICES
VOCATIONAL APTITUDE BATTERY (ASVAB)
-
AN OPTIMAL AND A HEURISTIC
APPROACH

by

Dietmar Kunde

March, 1997

Thesis Advisor: Robert F. Dell

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	Form Approved OMB No.0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March, 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE FORM ASSEMBLY FOR THE ARMED SERVICES VOCATIONAL APTITUDE BATTERY (ASVAB) - AN OPTIMAL AND A HEURISTIC APPROACH	5. FUNDING NUMBERS
6. AUTHOR(S) Dietmar Kunde	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT

The 1948 Selective Service Act established a process whereby all United States (US) military applicants take an aptitude test to measure their suitability for military job specialties. The latest version of these tests, the Armed Services Vocational Aptitude Battery (ASVAB), was introduced in 1968. Approximately 900,000 High School students from 14,000 US High Schools take the ASVAB test each year. This "paper and pencil" test requires the applicant to answer multiple choice questions (items) on a printed form. The creation of paper and pencil forms in one of the ten test topics is called form assembly. Form assembly consists of picking 20 to 35 items from an item pool of about 300 items such that: 1) each item appears on at most one form; 2) each form's result represents the applicant's capability; and 3) each form has the same level of difficulty. The thesis models the creation of paper and pencil forms as a mixed integer linear goal program and solves the problem both optimally and heuristically. Computational results for seven ASVAB-Tests show both methods help improve the form assembly process.

14. SUBJECT TERMS Optimization, Mixed Integer Linear Programming, Form Assembly, ASVAB	15. NUMBER OF PAGES 57
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

FORM ASSEMBLY FOR THE ARMED SERVICES VOCATIONAL APTITUDE BATTERY (ASVAB)
-
AN OPTIMAL AND A HEURISTIC APPROACH

Dietmar Kunde
Captain, German Army
M.S., Federal Armed Forces University, Munich, 1985

Submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL

March 1997

ABSTRACT

The 1948 Selective Service Act established a process whereby all United States (US) military applicants take an aptitude test to measure their suitability for military job specialties. The latest version of these tests, the Armed Services Vocational Aptitude Battery (ASVAB), was introduced in 1968. Approximately 900,000 High School students from 14,000 US High Schools take the ASVAB test each year. This "paper and pencil" test requires the applicant to answer multiple choice questions (items) on a printed form. The creation of paper and pencil forms in one of the ten test topics is called form assembly. Form assembly consists of picking 20 to 35 items from an item pool of about 300 items such that: 1) each item appears on at most one form; 2) each form's result represents the applicant's capability; and 3) each form has the same level of difficulty. The thesis models the creation of paper and pencil forms as a mixed integer linear goal program and solves the problem both optimally and heuristically. Computational results for seven ASVAB-Tests show both methods help improve the form assembly process.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The 1948 Selective Service Act established a process whereby all United States military applicants take an aptitude test to measure their suitability for military job specialties. The latest version of these tests, the Armed Services Vocational Aptitude Battery (ASVAB), was introduced in 1968. Approximately 900,000 High School students from 14,000 US High Schools take the ASVAB test each year. This "paper and pencil" test requires the applicant to answer multiple choice questions (items) on a printed form. The Defense Manpower Data Center, as an executive agency for the ASVAB, is responsible for the design, development and creation of the tests. The creation of paper and pencil forms in one of the ten test topics is called form assembly. Form assembly consists of picking 20 to 35 items from an item pool of about 300 items such that: 1) each item appears on at most one form; 2) each form's result represents the applicant's capability; and 3) each form has the same level of difficulty. This thesis models the creation of paper and pencil forms as a mixed integer linear goal program. One approach solves the program using commercially available optimization software. A second approach uses a local search with random restart heuristic. Both approaches yield good solutions. Computational results for the seven ASVAB-Tests show that combining both methods can improve the form assembly process. The Defense Manpower Data Center benefits from these computational results.

I. INTRODUCTION

The 1948 Selective Service Act established a process whereby all United States (US) military applicants take an aptitude test to measure their suitability for military job specialties. The latest version of these tests, the Armed Services Vocational Aptitude Battery (ASVAB), was introduced in 1968. A US Air Force Human Resources Laboratory study in 1973 calculated cost avoidance from these tests at $76.8 million per year for enlisted technical training [US Air Force Human Resources Laboratory 1973].

The ASVAB is currently given in about 14,000 US High Schools to about 900,000 potential applicants each year [Defense Manpower Data Center 1992]. This "paper and pencil" test requires the applicant to answer multiple choice questions (items). Each question has one correct answer that must be selected, on average, from a total of four choices. The ASVAB test consists of ten different areas of expertise. The categories — which have between 20 and 35 specific items each — are Arithmetic Reasoning (AR), Auto and Shop (AS), Coding Speed (CS), Electronics Information (EI), General Science (GS), Mechanical Comprehension (MC), Mathematical Knowledge (MK), Numerical Operations (NO), Paragraph Comprehension (PC), and Word Knowledge (WK).

The model developed in this thesis addresses only seven of the ten tests. The seven tests selected for use in the model's development are selected because they are similarly structured. That is, these seven tests are configured in a manner which makes the choice of the next eligible item independent of the item chosen before. In other words, there is no dependency among items from the perspective of the form assembly process.

The creation of paper and pencil forms for each category is called "form assembly." Multiple forms must be created in each category so that all applicants are not tested using the same form. "Form assembly" consists of picking 20 to 35 items from a pool of about 300 items such that: 1) each item appears on at most one form; 2) each form's result represents the applicant's capability; and 3) each form has the same level of difficulty. The item pool itself can be split into several item groups, where each group, called a taxonomy, requires a certain number of items per form.

This thesis models the creation of paper and pencil forms as a mixed integer linear goal program and solves the problem both <u>optimally</u> and <u>heuristically</u>.

A. TEST THEORY BACKGROUND

The measurement of a person's ability or skill level (denoted Θ) is commonly discretized into 100 intervals, so that each level can be expressed as a percentage. These intervals are then called percentiles of the ability. The skill level distribution over the potential applicant population is approximately normal allowing percentiles to be ranked from -3σ to $+3\sigma$ around a mean. A reasonable assumption is that the probability p of answering an item correctly increases as the percentile increases with p approaching 1 as the percentile goes to $+3\sigma$. Hence, this probability can be represented by a logistic function, referred to as an item response curve. A common model [Lord 1980] uses a three-parameter logistic function like the one adapted from Lord and Novick [1968] (Figure 1) with

2

$$p(\Theta) = c + \frac{1 - c}{1 + e^{-1.7 \cdot a(\Theta - b)}} \quad .$$

Parameter *a* is a proportionality factor for the slope at the inflection point. It represents the discriminating power; in other words, how capable an item is to distinguish between applicants. Figure 2 shows an example where item 1 has a steeper curve in the percentile range (50,60) than item 2 and therefore provides greater discrimination between individuals at percentiles 50 and 60.
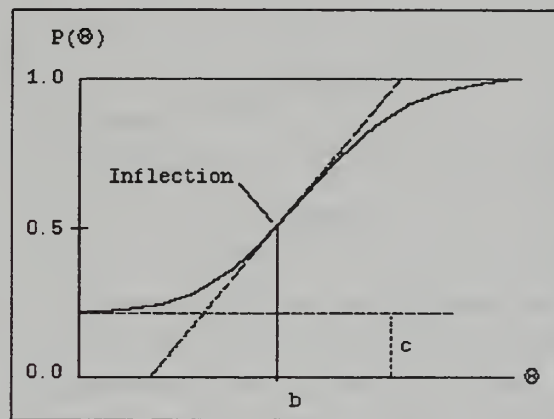

Figure 1: Parameters of the Logistic Function.
The logistic function represents the probability of answering an item correctly and is defined with parameters (*a*, *b* and *c*). Parameter *a* is proportional to the slope at the inflection point: slope = .425a(1-c). Parameter *b* indicates an item's difficulty level by defining the position of an item's curve along the ability scale θ. Parameter *c* indicates the guessing parameter [Lord 1980].

Parameter *b* indicates an item's difficulty level by defining the position of an item's curve along the ability scale θ (i.e., when the percentile Θ_i corresponding to the probability of a correct answer is 0.5).

Parameter *c* indicates the guessing parameter or the probability of answering an item correctly given an ability

Parameter *c* indicates the guessing parameter or the probability of answering an item correctly given an ability falling greater than 3σ below the mean [Lord 1980]. This guessing parameter does not necessarily reflect the probability to select one correct answer from a certain number of possible choices.
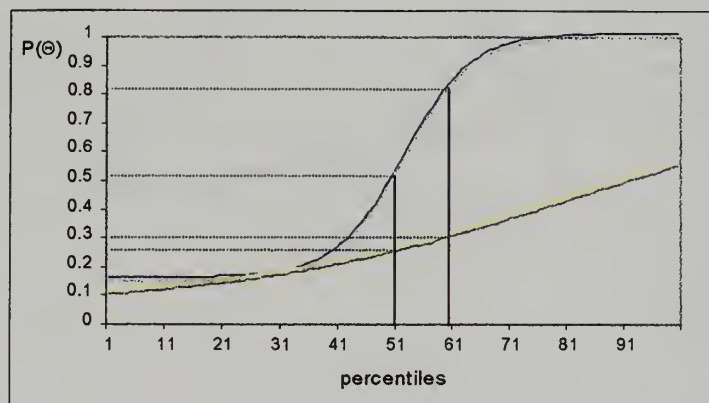

Figure 2: Example of the Discriminating Power.

Figure 2 provides an example of the discriminating power of two items for two applicants with percentiles 50 and 60. Item 1 has a steeper curve in the percentile range (50,60) than item 2 and therefore provides greater discrimination between individuals at percentiles 50 and 60.

In practice, 1,000 to 10,000 applicants pretest an item and the parameters *a*, *b* and *c* are estimated from the results. From the item response curve, an item information curve is determined (Figure 3). The item information curve describes the potential information contribution of an item to a test form at each percentile. These item information curves comprise the bulk of the data for this thesis.

These item information curves are independent and additive when it is assumed that the information contribution of an item to the whole form does not depend on other items included on the form [Lord 1980]. Therefore all of a form's

item information curves can be added to get an overall information curve. This overall information curve is commonly denoted as the precision of the form.
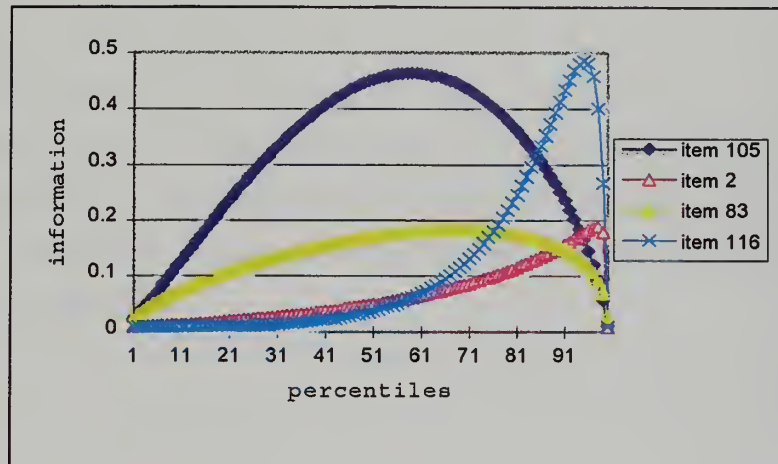

Figure 3: Item Information Curves.
Figure 3 displays examples of different item information curves. These curves describe the potential information contribution of an item to a test form at each percentile.

Empirical research and testing has produced a "reference curve" for each test representing the desired information distribution over a form's percentiles. Since the establishment of a standard reference curve in 1980, some item pools have changed and it is now possible to provide forms with "better" information curves than the reference curve. In such cases, these curves are the new desired information distribution but cannot be called reference curves for historic purity. Regardless, in this thesis, we refer to the preferred curve as the "goal curve."

B. OUTLINE

Chapter II provides information about research related to this thesis. Chapter III formulates the form assembly process as a mixed integer linear goal programming problem and discusses a heuristic to solve it. Chapter IV provides results obtained from solving the formulation using a heuristic and the General Algebraic Modeling System (GAMS) [Brooke, Kendrick and Meeraus 1992] with the solver OSL [GAMS 1995]. Chapter V compares the two solution methods and presents conclusions.

II. RELATED RESEARCH

The bulk of the literature on aptitude and ability tests involves the concept of item validity [Lord 1980]. Validity in this case is taken to be the extent to which a test score actually predicts future performance. Toquarn, Corpe and Dunette [1991] review more than 10,000 articles related to validity as it pertains to ability tests. Their literature review highlights the significant effort associated with this issue. As pertains specifically to the ASVAB, Maier and Truss [1985] give an example of that test's predictability. In this study, the authors demonstrate that performance on the ASVAB tests is statistically related to training outcome measures of various US Marine Corps technical schools.

The present study uses data provided by the Defense Manpower Data Center (DMDC). Again, as explained on page four, these data consist of roughly 300 item information curves, each curve derived by standard statistical procedures [Lord and Novick 1968] from item response curves. These data are assumed to be representative with respect to the validity issue. Accordingly, the DMDC data used in the present study are used simply to demonstrate a methodological approach to "form assembly." They are not being used to demonstrate their predictive validity.

Unlike the validity literature, there exist only a few publications addressing <u>assembly</u> or construction of ability or aptitude tests. Berger, Gupta and Berger [1988] present the construction of Form P for the Air Force Officer Qualifying Test (AFOQT). They develop two forms of the test by adding new items to an old form. The objective is to construct two new forms which are equivalent and parallel to

the original form. "Equivalence" means that each form has the same information content. "Parallel" means that the outcome of the test is independent of the form the applicant has taken. Their approach is <u>heuristic</u>. The heuristic is straight forward. They select items with the most discriminating power from the old form; check them against new items; and replace old items with new items that provide the best match; that is, a match which produces the smallest information differences between the old and the new form.

Baker and Wall [1996] use a form assembly similar to the heuristic approach presented in this thesis. They focus on a statistical analysis of the Interest Finder Test, a test to help students explore their occupational and career interests [DMDC 1992]. They describe form assembly as consisting of two stages. The first stage screens the item pool and the second stage uses a heuristic algorithm to assign items to the form. Their heuristic selects an initial group of items and exchanges items when replacement considerations improve the form. The objective function is a weighted function that minimizes statistical differences between the current form and a desired form. These statistical differences are essentially the mean and standard deviation of scaling parameters for the test. The actual criteria for the initial item selection and results with respect to form assembly are beyond the scope of this paper.

In summary, the literature review did not reveal prior attempts to use optimization in form assembly and only provided scant references to the use of heuristic approaches. The next chapter discusses the optimization and heuristic approaches.

III. OPTIMIZATION MODEL AND HEURISTIC

A. OPTIMIZATION MODEL

The form assembly problem can be formulated as a mixed integer linear goal programming problem (see Charnes and Cooper [1961] for a discussion of goal programming) consisting of two goals. One goal is to assemble forms so each form's information curve is as close as possible to the goal curve. The second goal is to make each form's information curve as "parallel" as possible to one another. The "parallel" goal seeks an exam, where results are independent of the form the applicant has taken. An exam with all forms exactly matching the goal curve would simultaneously satisfy both goals but this is typically not possible. The parallel goal therefore encourages each form to be close to the goal curve.

We implement the first goal by allowing the deviation from the reference curve to vary in groups where deviation within the group has the same penalty per unit and groups closer to the goal curve have a smaller penalty per unit. Figure 4 provides an example of the penalty groups. Any vertical deviation between the goal curve and form curve is penalized.

information | **percentiles** | **68**

goal curve

Group$_1$

Group$_2$

ΔI

Group$_3$

form curve
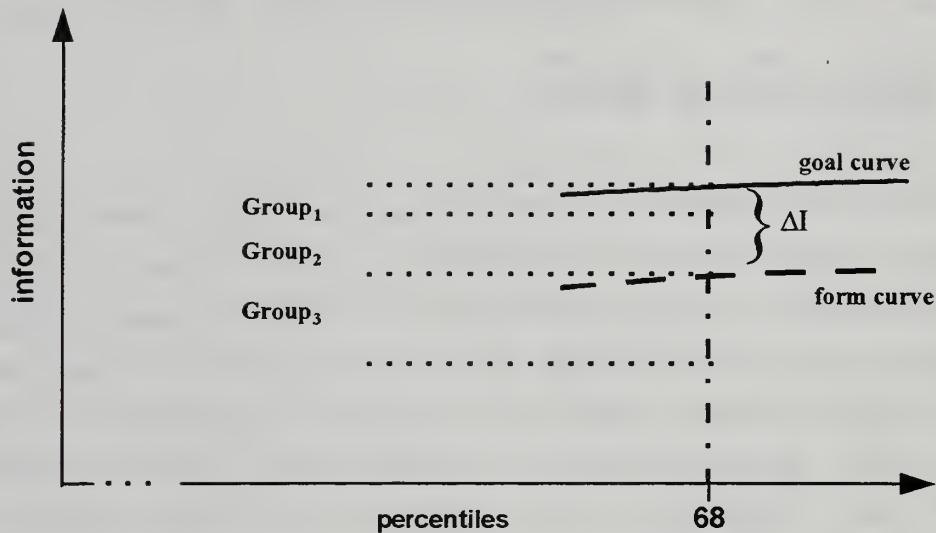
Figure 4: Penalty Groups.

This figure displays at percentile 68 how deviation from the goal curve can be measured in different groups. The vertical distance ΔI would be penalized per unit with the penalty for group 1 for those units of ΔI within group 1 and with the penalty per unit for group 2 for those units of ΔI within group 2. Since it is desired to be as close to the goal curve as possible, group 1's penalty per unit would be less than group 2's penalty per unit.

The formulation follows.

Indices:

    i : item from the item pool;

    p : percentile(ability level);

    f : form to be assembled $(1,2,..,F)$;

    t : taxonomy$(1,2,..,T)$; and

    g : penalty group.

Data:

    CAT_g the maximum deviation between a form and the goal curve in group g;

    INF_{ip} information value of item i at percentile p;

10

NITEM$_t$ the required number of items in taxonomy t;

PARAWEI   weight that combines the two goals;

PENALTY$_g$ penalty per unit deviation within group g; and

SHAPE$_p$ the information value for the goal curve at percentile p.

Variables:

x_{if} 1, if item i is used on form f;

py_{pfg} deviation above the desired shape in group g at percentile p on form f;

ny_{pfg} deviation below the desired shape in group g at percentile p on form f;

Delplus$_f$ the total information form 1 contains that exceeds form f; and

Delneg$_f$ the total information form f contains that exceeds form 1.

Formulation:

$$\min \quad \sum_p \sum_f \sum_g PENALTY_g \cdot (py_{pfg} + ny_{pfg})$$

$$+ \quad PARAWEI \cdot \sum_{f>1} (Delplus_f + Delneg_f) \qquad (1)$$

$$\sum_g py_{pfg} \geq \sum_i INF_{ip} \cdot x_{if} - SHAPE_p \qquad \forall p, f \quad (2)$$

$$\sum_g ny_{pfg} \geq - \sum_i INF_{ip} \cdot x_{if} + SHAPE_p \qquad \forall p, f \quad (3)$$

$$\sum_i x_{if} = NITEM_t \qquad \forall f, t \quad (4)$$

$$\sum_f x_{if} \le 1 \qquad\qquad\qquad \forall i \qquad (5)$$

$$\sum_i \sum_p INF_{ip} \cdot x_{i1} \; - \; \sum_i \sum_p INF_{ip} \cdot x_{if} \qquad\qquad \forall f > 1 \qquad (6)$$
$$= \; Delplus_f \; - \; Delneg_f$$

$$0 \le py_{pfg} \le CAT_g \qquad\qquad\qquad \forall p, f, g \quad (7)$$

$$0 \le ny_{pfg} \le CAT_g \qquad\qquad\qquad \forall p, f, g \quad (8)$$

$$x_{if} \; binary \qquad\qquad\qquad\qquad \forall i, f$$

$$Delplus_f, \; Delneg_f \ge 0 \qquad\qquad\qquad \forall f .$$

The first component of the objective function,
$$\sum_p \sum_f \sum_g PENALTY_g \cdot (py_{pfg} + ny_{pfg}) \; ,$$
minimizes the vertical distances (weighted deviation) between the goal curve and the assembled forms. The second component,
$$PARAWEI \cdot \sum_f (Delplus_f + Delneg_f) \; ,$$
encourages forms to have the same information. A second component having value zero does not necessarily imply parallel forms since the vertical distances at percentile p from form 1 to form f can have positive or negative signs depending on whether form f is above or below form 1. These positive and negative distances can sum up to zero producing two forms where $Delplus_f = Delneg_f = 0$. Nevertheless, the second component has empirically produced parallel forms and requires only F-1 additional constraints. Constraints (2) and (3) determine the positive and negative deviation at each percentile between the assembled forms and the goal curve. Constraint (4) ensures the required number of items

per taxonomy is satisfied. Constraint (5) ensures that each item is used at most once. Constraint (6) determines the total information difference between form 1 and other forms. Constraints (7) and (8) bound the positive and negative deviations.

B. HEURISTIC APPROACH

Solving the previous problem optimally has taken extensive computation time as shown in the next chapter. To provide solutions quickly a local search with random restart heuristic (e.g., [Papadimitriou and Steiglitz 1982]) is developed.

The main objectives for the heuristic are to quickly complete one assembly and to quickly evaluate small variations to the assembly. The heuristic uses only integer arithmetic within efficient code to help improve performance.

The heuristic starts by dividing the item pool into arrays of items where each array corresponds to a taxonomy. These sub-item pools are eligible sets (ES_t) for each taxonomy.

Each form consists of vectors for each taxonomy ($Assign_{tf}$). The algorithm consists of three main procedures (Figure 5): *fill_initial_form; do_swap; and improve_parallel*.

Figure 6 displays the pseudocode for the procedure *fill_initial_forms*. A random number generator [Lewis, Goodmann and Miller 1969] is used to assemble the initial forms subject to all constraints.
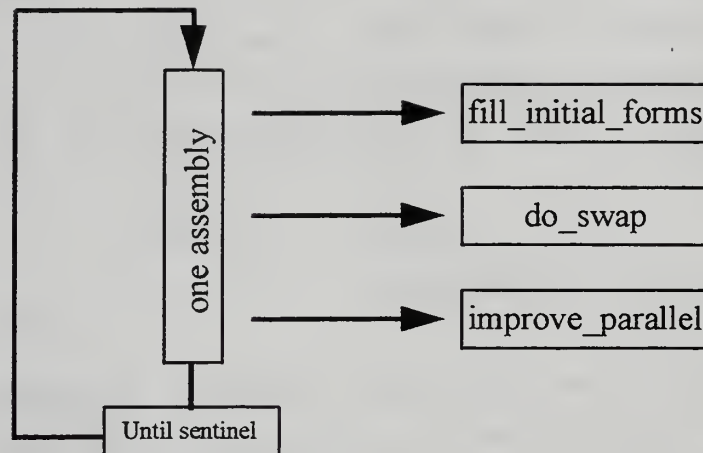
Figure 5: Main Procedures of the Heuristic.
This figure shows the main procedures for the heuristic algorithm. A loop over one assembly of all forms runs as often as the user has chosen. The best assembly is the result.

```
 1 Assign_tf ←∅; initialize ES_t (assume |ES_t|≥F*NITEM_t}
 2 for f = 1 to F
 3    for t = 1 to T
 4       while |Assign_tf| < NITEM_t
 5          randomly select item i from ES_t
 6          Assign_tf ← Assign_tf ∪ {i}
 7          ES_t ← ES_t - {i}
 8       end
 9    end
10 end
```

Figure 6: The Pseudocode for the Procedure *fill_initial_forms*.
This figure shows how the heuristic randomly assembles the initial forms. The indices and variables match those from the optimization model. $Assign_{tf}$ contains items on form f in taxonomy t. ES_t contains all items in taxonomy t not currently used on any form.

The procedure *do_swap* defines a swap as the exchange of an item from a form ($i_{out} \in Assign_{tf}$) with an item from the

appropriate eligible set ($i_{in} \in ES_t$). Figure 7 shows the pseudocode for this procedure.

```
 1 improve ← 1
 2 while improve > 0
 3    improve ← 0
 4    for t = 1 to T
 5       for f = 1 to F
 6          for each item i_out ∈ Assign_tf
 7             sofar ← ObjFctValue_old
 8             Assign_tf ← Assign_tf - {i_out}
 9             for each item (i_in) ∈ ES_t
10                Assign_tf ← Assign_tf + {i_in}
11                calculate ObjFctVal_new
12                   if ObjFctVal_new_f < sofar (improvement)
13                      sofar ← ObjFctVal_new
14                      candidate = i_in
15                   end if
16                Assign_tf ←  Assign_tf - {i_in}
17             end
18             if sofar < ObjFctValue_old
19                swap candidate with i_out
20                update involved curves
21                improve ← improve +1
22             end if
23          end
24       end
25    end
26 end while
```

Figure 7: The Pseudocode for the Procedure *do_swap*.
This figure shows how items swapping improves forms. ObjFctValue_old is the sum of all deviation between form f and the goal curve before potentially swapping an item and ObjFct_new is after a potential swap. The procedure repeats until no swap yields a decrease to the objective function of any form.

The objective function value measuring the effectiveness of the swap is the sum of all deviations between form f and the

goal curve. Improvement, as it is used in this context means a decrease of the objective function value, caused by swapping an item. This procedure runs through all forms and eligible sets and checks whether a swap yields improvement. The while-loop repeats as long as at least one improvement is found across all forms and eligible sets.

To increase the speed of the algorithm a baseline for checking the swaps is used. A baseline in this context is the sum of all item information curves currently assembled without the item considered for exchange (i_{out}). Within the pseudocode of Figure 7, the baseline can be calculated after step 8; and doing so reduces the computational effort needed to determine the new objective function value in step 11. Only the 100 information values of item i_{in} have to be added to the baseline instead of summing over all items currently assigned. The swap is executed after all items of the eligible set have been examined with that item that gives the most improvement (candidate).

The procedure *improve_parallel* checks if swapping items between forms can improve the forms. The procedure starts by finding the form with the smallest sum of all deviations from the goal curve sofar. This best form is the one with which the other forms have to be aligned. Figure 8 displays the pseudocode for the procedure *improve_parallel*. At this stadium, the heuristic does not allow the objective function to increase.

An improving swap between forms happens only after all items within a taxonomy on all forms have been compared with an item on the best form. The calculation of the curves uses the baseline principle again. *Improve_parallel* terminates when no item is swapped on any form.

```
 1 improve ← 1
 2 while improve > 0
 3    improve ← 0
 4    find best form f_best
 5    for t = 1 to T
 6      for each item $i_{out}$ ∈ $Assign_{tf\_best}$
 7        $Assign_{tf\_best}$ ← $Assign_{tf\_best}$ - {$i_{out}$}
 8        for f = 1 to F excluding f_best
 9          sofar ← $(ObjFctValue_f + ObjFctValue_{f\_best})_{old}$
10          for each item $i_{in}$ ∈ $Assign_{tf}$
11            $Assign_{tf\_best}$ ← $Assign_{tf\_best}$ + {$i_{in}$}
12            $Assign_{tf}$ ← $Assign_{tf}$ - {$i_{in}$} + {$i_{out}$}
13            calculate ObjFctValues
14            better? ← $(ObjFctValue_f + ObjFctValue_{f\_best})_{new}$
15              if better? < sofar then improvement
16                  sofar ← better?
17                  candidatein  = $i_{in}$
18                  candidateout = $i_{out}$
19             end if
20            $Assign_{tf}$ ← $Assign_{tf}$ + {$i_{in}$} - {$i_{out}$}
21            $Assign_{tf\_best}$ ← $Assign_{tf\_best}$ - {$i_{in}$}
22          end
23        end
24        if sofar < $(ObjFctValue_f + ObjFctValue_{f\_best})_{old}$
25            swap candidates
26            update involved curves
27            improve ← improve +1
28        end if
29      end
30    end
31 end while
```

Figure 8: The Pseudocode for the Procedure *improve_parallel*.
This figure shows swaps allowed between forms. A swap, given it improves
the objective function value, occurs after one item on the best form has
been compared with all other assigned items on the other forms.

17

IV. COMPUTATIONAL RESULTS

The task is to assemble forms for seven different tests: Arithmetic Reasoning (AR), Auto and Shop (AS), Electronics Information (EI), General Science (GS), Mechanical Comprehension (MC), Mathematical Knowledge (MK), and Word Knowledge (WK). Table 1 lists the test specifications.

Test	Item Pool size	Forms needed	Items on form	Taxonomies
AR	338	2	30	5
AS	196	2	25	2
EI	190	2	20	4
GS	313	2	25	12
MC	296	4	25	6
MK	327	4	25	5
WK	276	2	35	2

Table 1: Test Requirements and Item Pools.
This table lists the specifications for each of the tests. For example, the AR-Test requires the creation of two forms each having 30 items. The 30 items, falling into five taxonomies, must be selected from an item pool of 338 items.

A. OPTIMIZATION PARAMETER SETTINGS

The optimization model formulated in the previous chapter requires the specification of a number of parameters. A summary sheet for each test contains results as well as parameter settings. We use the AR-Test as an example.

Figure 9 shows the implemented objective function. All values were empirically developed. The penalties for the unbounded variables, py4 and ny4, are 100. Other values are: CAT_1 = 0.01; CAT_2 = 0.05; CAT_3 = 0.10; $penalty_1$ = 0.00001; $penalty_2$ = 1.00; $penalty_3$ = 5.00; and PARAWEI = 25.

$$
\begin{aligned}
(\sum_f \sum_p & \ 100 \cdot py4_{pf} \ + \ 100 \cdot ny4_{pf} \\
+ & \ 0.00001 \cdot py1_{pf} + 1 \cdot py2_{pf} + 5 \cdot py3_{pf} \\
+ & \ 0.00001 \cdot ny1_{pf} + 1 \cdot ny2_{pf} + 5 \cdot ny3_{pf}) \\
+ & \ 25 \ * \ \sum_f (Delplus_f \ - \ Delneg_f)
\end{aligned}
$$

Figure 9: The objective function parameters for the optimization model. This figure shows the objective function implemented in GAMS for the AR-Test. It measures the overall distance between the forms and the goal curve at each percentile. The pys and nys are the deviation variables. 25 * Σ(Delplus - Delneg) is the subgoal to encourage parallel forms.

We use only upper bounds on the deviation variables (CAT_g) for groups 1, 2 and 3. The following pages display for each test the bounds for the penalty groups and the weights for the subgoal.

B.   OPTIMIZATION RESULTS

This section shows results for the assembled tests. The integrality gap provided is the difference between the best integer solution identified and a lower bound on the solution, expressed as a percentage of the lower bound. The results for all tests are presented in alphabetical order.

Table 2 summarizes the numerical results obtained. Figures 10 to 16 show graphical results.

Test	objfctvalue lower bound (❶)	objfctval best solution (❷)	integrality gap (%) (❸)	runtime (seconds)
AR	865.97	932.33	7.6	15,260
AS	2,788.00	2,862.73	2.7	215
EI	9,489.65	9,561.94	1.0	17
GS	8,095.66	8,433.03	4.2	312
MC	125.04	1,187.11	850.0	50,000
MK	2,006.71	7,278.24	260.0	50,000
WK	3,588.31	5,188.42	39.2	13,934

Table 2: Numerical Results of the Optimization Assembly.
Table 2 summarizes all numerical results for tests assembled using optimization, where objfctvalue = Objective Function Value. The integrality gap provided is the difference between the best integer solution identified and a lower bound on the solution, expressed as a percentage of the lower bound (e.g., ❸=(❷-❶)/❶).

Model results come from an IBM RS6000 Model 590 workstation using GAMS and the OSL solver. The model size varies, primarily according to the number of forms and the cardinality of the item pool. The approximate size of the largest model, MK-Test, is shown below:

number of constraints: 1,150;
number of continuous variables: 4,500;
number of binary variables: 1,300; and
number of non-zero elements: 250,000.

AR - Test (Arithmetic Reasoning):

General Requirements:

    forms: 2;

    items: 30 each; and

    taxonomies: 5 (7,8,5,5,5 items in taxonomy 1 to 5).

Settings:

    CAT-values: 0.01, 0.05, 0.1;

    penalties: 0.00001, 1, 5;

    PARAWEI: 25; and

    item pool: 338 items.

Numerical Results:

    objective function value (lower bound):   865.97;

    objective function value (best solution): 932.33;

    integrality gap: 7.6 %; and

    runtime (seconds): 15,260 (4.2 hours).

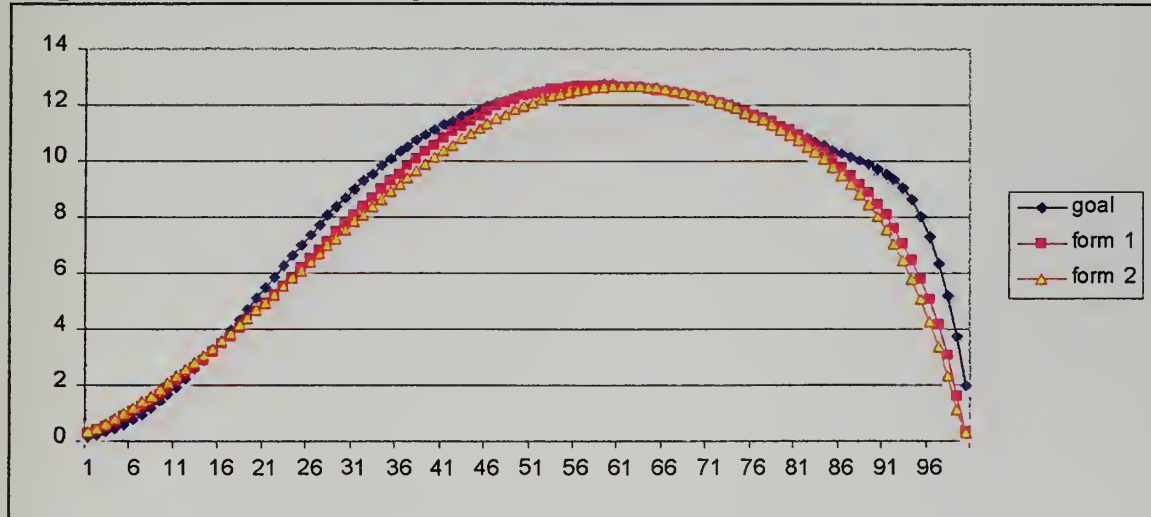Graphical Results: Figure 10 below.
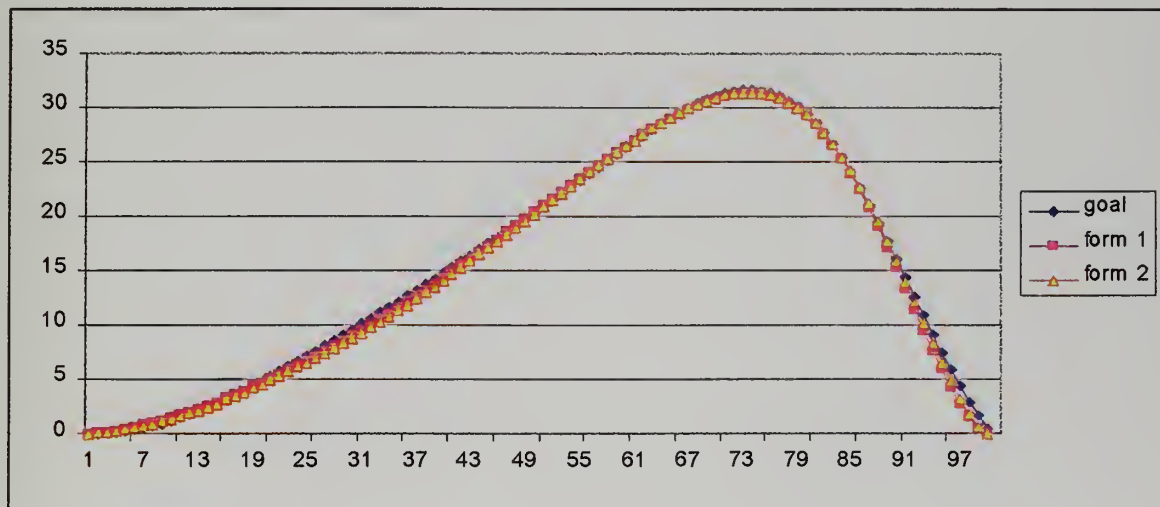
Figure 10: Graphical Results for the AR-Test.

This figure shows results obtained for the AR-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

AS - Test (Auto and Shop):

General Requirements:

    forms: 2;

    items: 25 each; and

    taxonomies: 2 (11, 13 items in taxonomy 1 and 2).

Settings:

    CAT-values: 0.05, 0.1, 0.5;

    penalties: 0.00001, 1, 5;

    PARAWEI: 25; and

    item pool: 196 items.

Numerical Results:

    objective function value (lower bound): 2,788.00;

    objective function value (best solution): 2,862.73;

    integrality gap: 2.7 %; and

    runtime (seconds): 215.

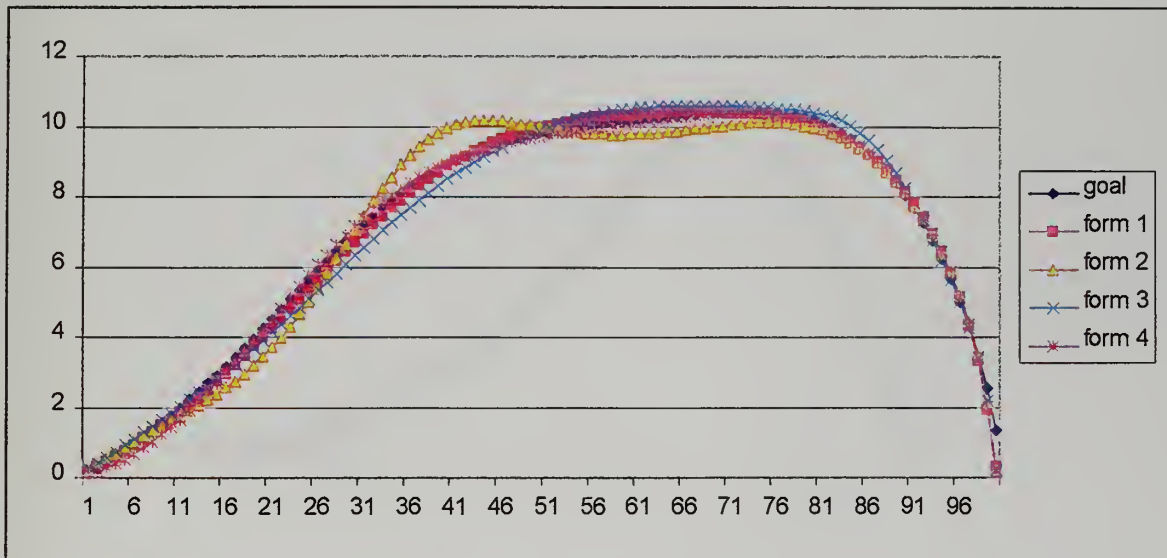Graphical Results: Figure 11 below.
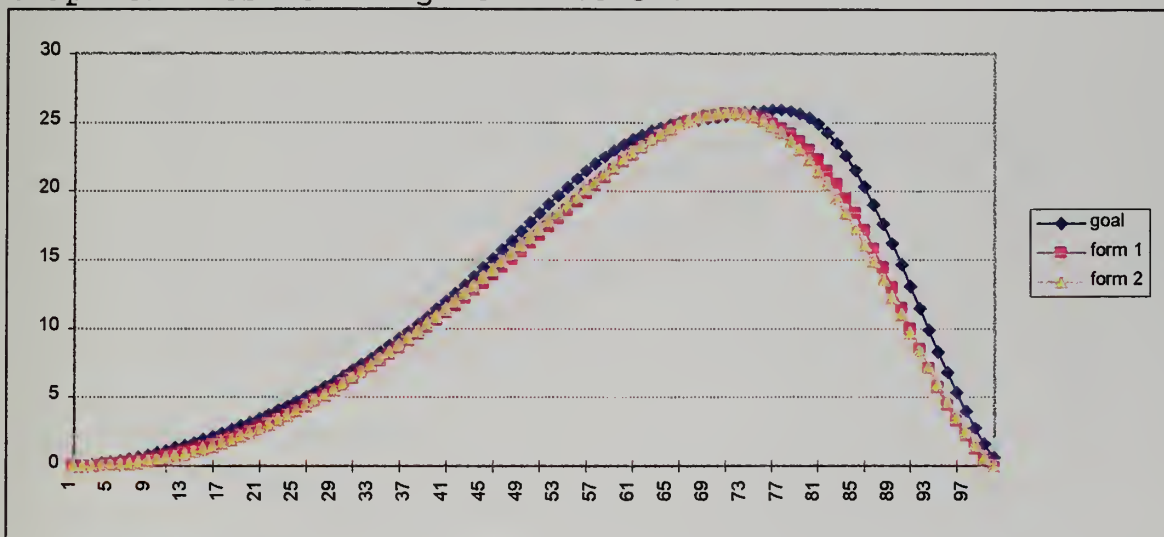
Figure 11: Graphical Results for the AS-Test.

This figure shows results obtained for the AS-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

EI - Test (Electronics Information):

General Requirements:

    forms: 2;

    items: 20 each; and

    taxonomies: 4 (10,4,2,4 items in taxonomy 1 to 4).

Settings:

    CAT-values: 0.05, 0.1, 0.7;

    penalties: 0.00001, 1, 10;

    PARAWEI: 3; and

    item pool: 190 items.

Numerical Results:

    objective function value (lower bound): 9,489.65;

    objective function value (best solution): 9,561.94;

    integrality gap: 1.0 %; and

    runtime (seconds): 17.
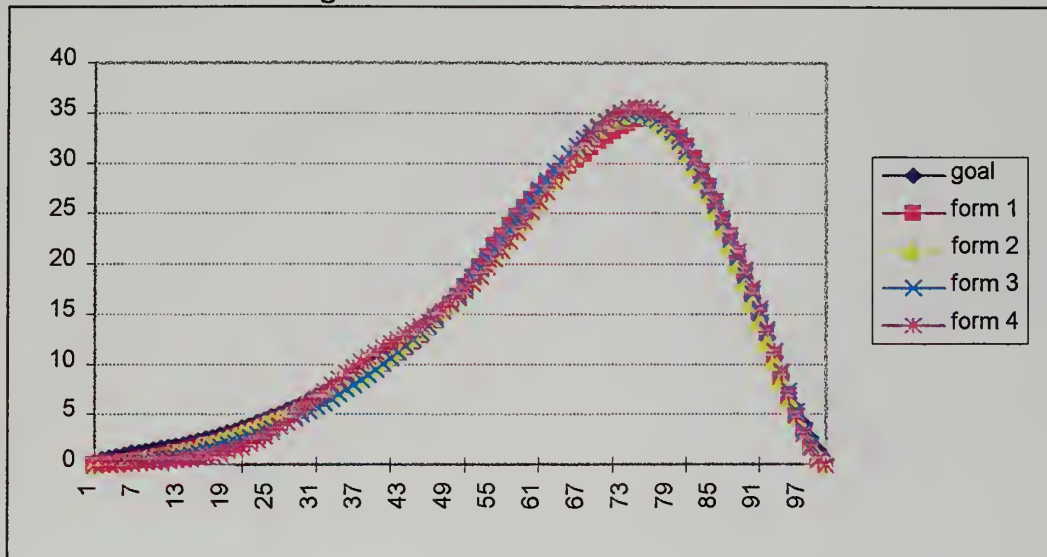
Graphical Results: Figure 12 below.
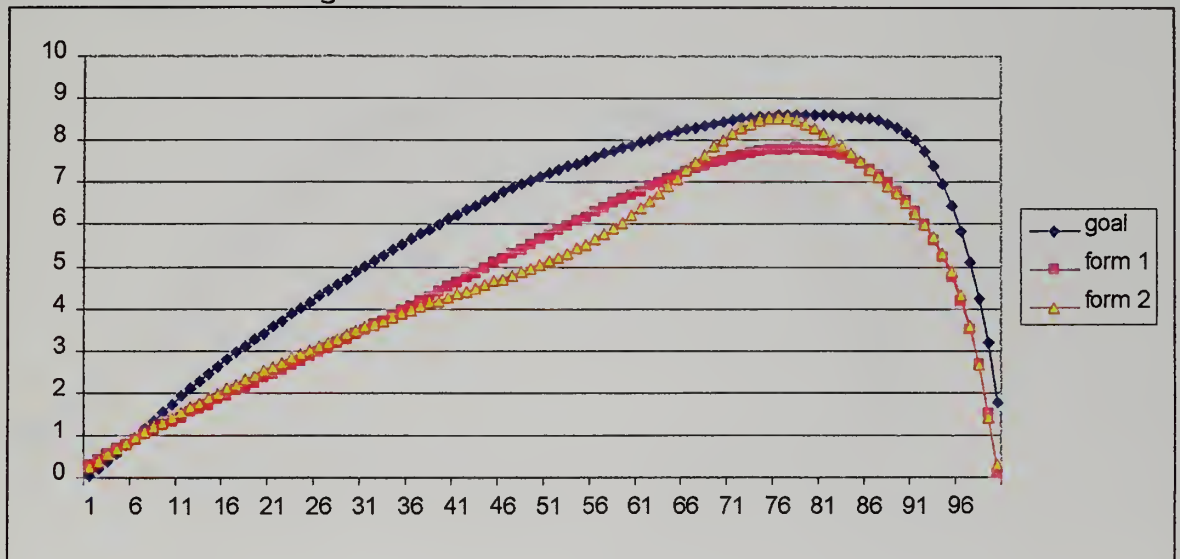
Figure 12: Graphical Results for the EI-Test.

This figure shows results obtained for the EI-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

GS - Test (General Science):

General Requirements:

    forms: 2;

    items: 25 each; and

taxonomies: 12 (3,3,4,2,2,3,1,2,2,1,1,1).

Settings:

    CAT-values: 0.05, 0.1, 0.5;

    penalties: 1, 10, 100;

    PARAWEI: 100; and

    item pool: 313 items.

Numerical Results:

    objective function value (lower bound): 8,095.66;

    objective function value (best solution): 8,433.03;

    integrality gap: 4.2 %; and

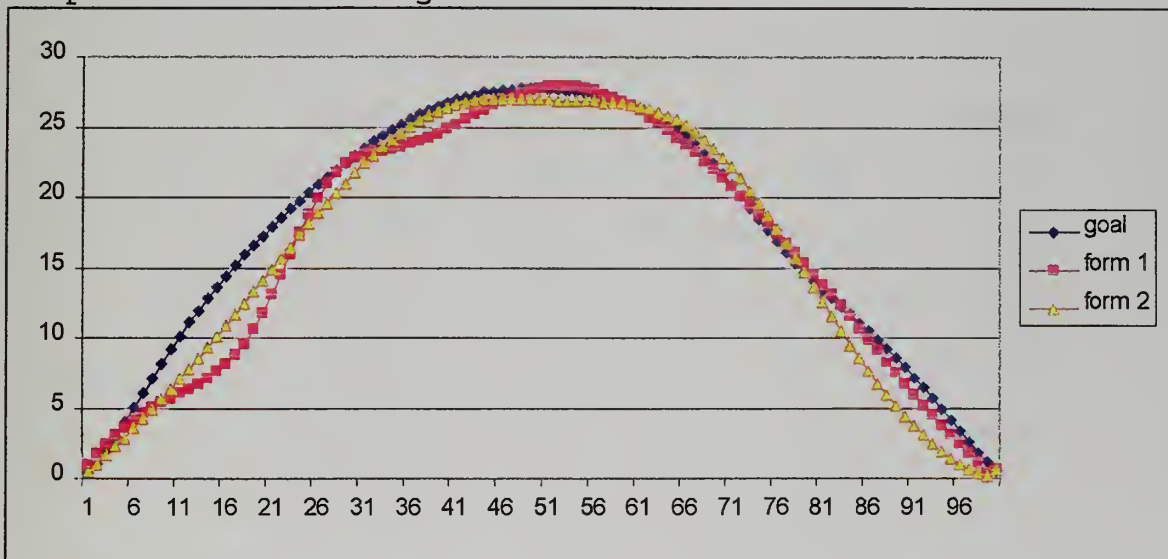   runtime (seconds): 312.

Graphical Results: Figure 13 below.

Figure 13: Graphical Results for GS-Test.

This figure shows results obtained for the GS-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

MC - Test (Mechanical Comprehension):

General Requirements:

    forms: 2;

    items: 25 each; and

    taxonomies: 6 (11,2,2,2,4,4 items in taxonomy 1 to 6).

Settings:

    CAT-values: 0.01, 0.05, 0.1;

    penalties: 0.00001, 1, 5;

    PARAWEI: 300; and

    item pool: 296 items.

Numerical Results:

    objective function value (lower bound):   125.04;

    objective function value (best solution): 1,187.83;

    integrality gap: 850 %; and

    runtime (seconds): 50,000 (13.8 hours).

Graphical Results: Figure 14 below.

Figure 14: Graphical Results for the MC-Test.

This figure shows results obtained for the MC-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

MK - Test (Mathematical Knowledge):

General Requirements:

    forms: 4;

    items: 25 each; and

    taxonomies: 5 (3,5,9,7,1 items in taxonomy 1 to 5).

Settings:

    CAT-values: 0.05, 0.1, 0.5;

    penalties: 1, 10, 100;

    PARAWEI: 300; and

    item pool: 327 items.

Numerical Results:

    objective function value (lower bound): 2,006.71;

    objective function value (best solution): 7,278.24;

    integrality gap: 7.3 %; and

    runtime (seconds): 50,000 (13.8 hours).

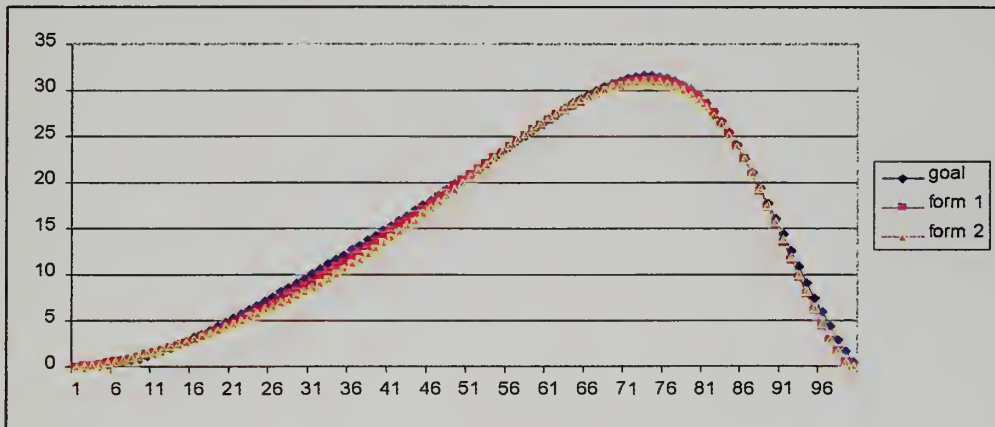Graphical Results: Figure 15 below.

Figure 15: Graphical Results for the MK-Test.

This figure shows results obtained for the MK-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 to form 4 are the information curves for each form.

WK - Test (Word Knowledge):

General Requirements:

    forms: 2;

    items: 25 each; and

    taxonomies: 2 (13,22 items in taxonomy 1 and 2).

Settings:

    CAT-values: 0.01, 0.05, 0.1;

    penalties: 0.000001, 1, 5;

    PARAWEI: 500; and

    item pool: 276 items.

Numerical Results:

    objective function value (lower bound): 3,588.32;

    objective function value (best solution): 5,188.42;

    integrality gap: 39.2 %; and

    runtime (seconds): 13,934 (3.9 hours)

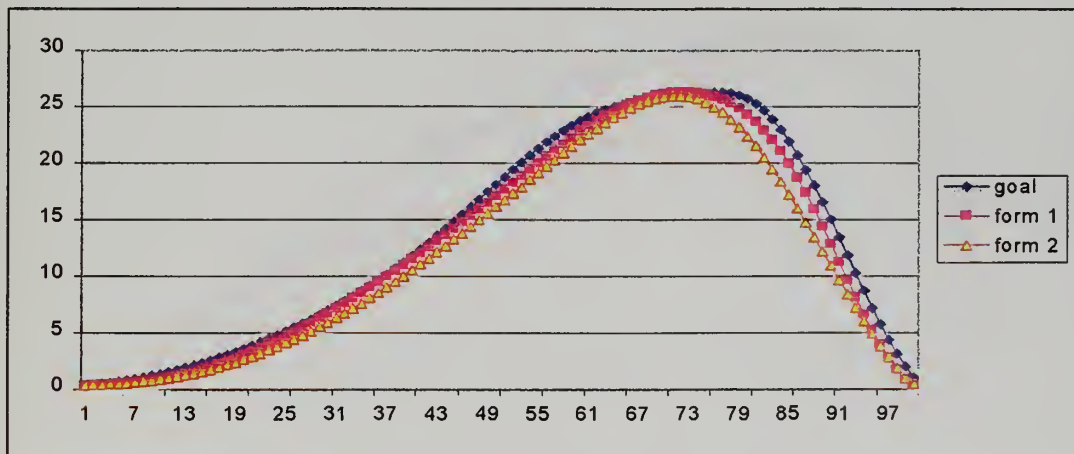Graphical Results: Figure 16 below.

Figure 16: Graphical Results for the WK-Test.

This figure shows results obtained for the WK-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

C. RESULTS OF THE HEURISTIC APPROACH

The objective function implemented in the heuristic is as follows:

$$\sum_p \sum_f (py_{pf} + ny_{pf})$$

This simplification of the objective function previously used (i.e., unweighted deviations and no parallel subgoal) was chosen for ease of computation.

The following pages display the objective function values per repetition (random restart) of the heuristic as well as the graph for the best solution found (Figures 17 to 30).

The heuristic algorithm is implemented on a Pentium 166 PC, written in Standard Pascal [e.g., Silicon Valley Software 1991]. Table 3 shows the runtimes and the objective function values.

Test	Objective function value	Repetitions	Runtime (seconds)
AR	97.74	100	120
AS	230.77	100	150
EI	227.10	100	120
GS	117.13	100	130
MC	47.80	100	250
MK	257.94	100	280
WK	280.68	100	160

Table 3 : Results for tests assembled with the Heuristic Approach. As the runtimes show, the heuristic provides results very quickly.

29

AR - Test (Arithmetic Reasoning):

General Requirements:

     forms: 2;

     items: 30 each; and

     taxonomies: 5 (7,8,5,5,5 items in taxonomy 1 to 5);

Execution Specifics:

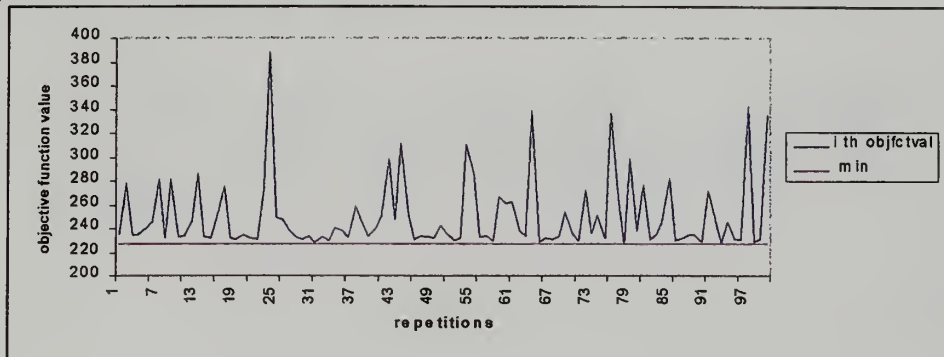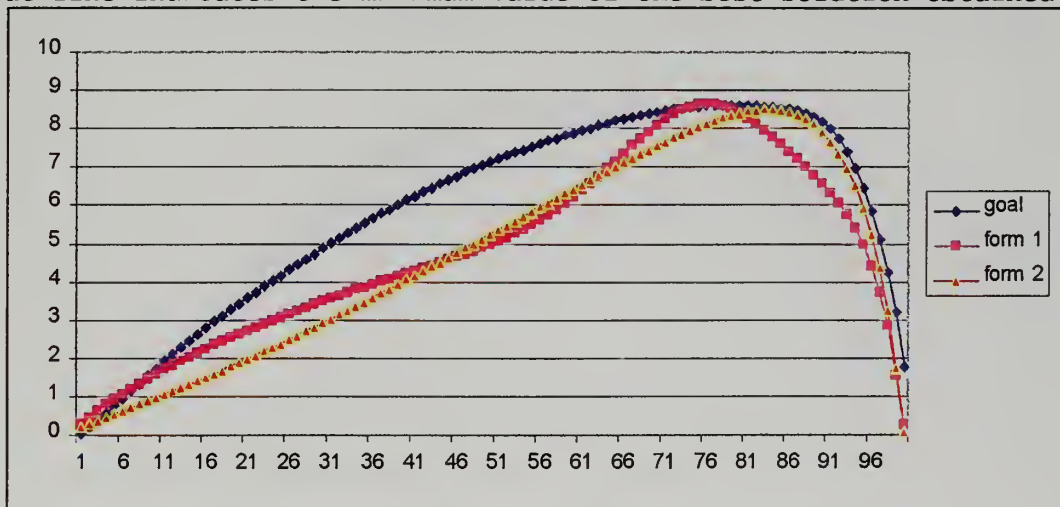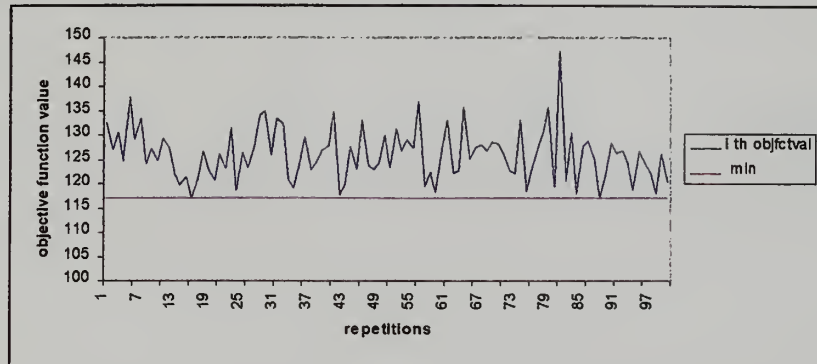   repetitions: 100; and

     objective function value: 97.74.
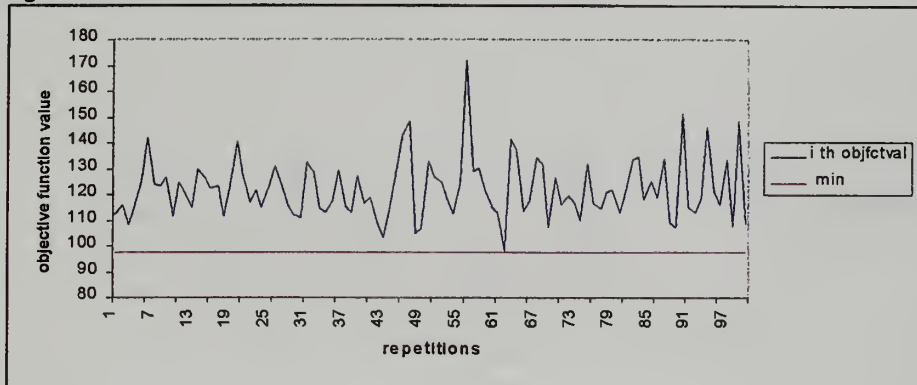
Figure 17: Objective Function Values for each Random Restart. The flat line indicates the minimum value.
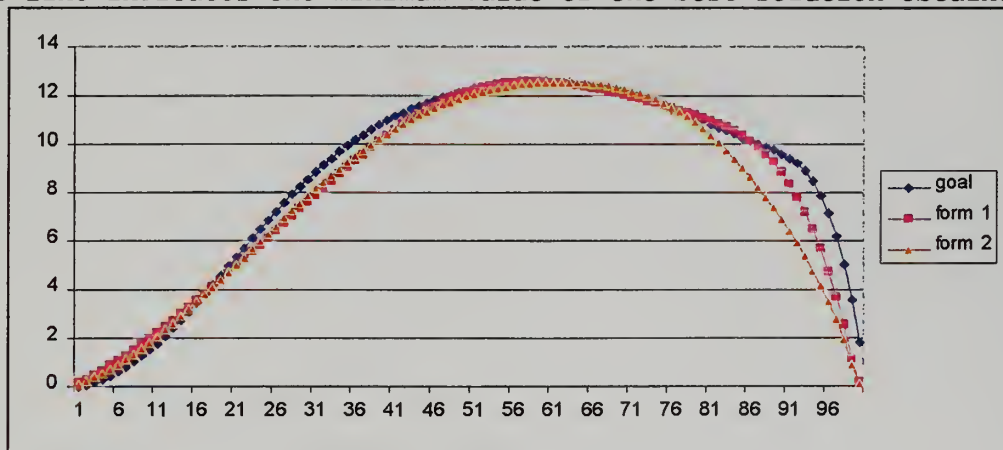

Figure 18: Graphical Results for the AR-Test.
This figure shows results obtained for the AR-Test with information on the vertical axis and the percentiles on the horizontal axis. Form 1 and form 2 are the information curves for each form.

AS - Test (Auto and Shop):

General Requirements:

    forms: 2;

    items: 25 each; and

    taxonomies: 2 (11, 13 items in taxonomy 1 and 2).

Execution Specifics:

    repetitions: 100; and

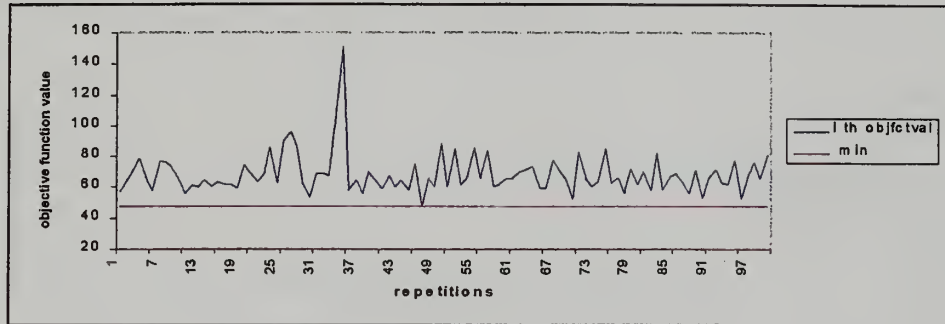   objective function value: 230.77.
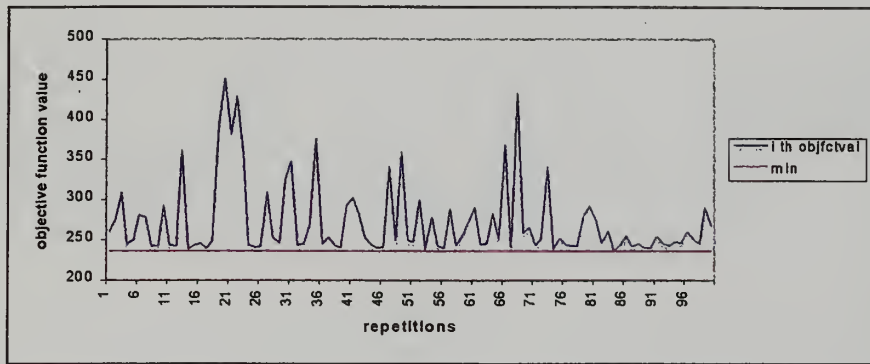
Figure 19: Objective function values for each Random Restart.
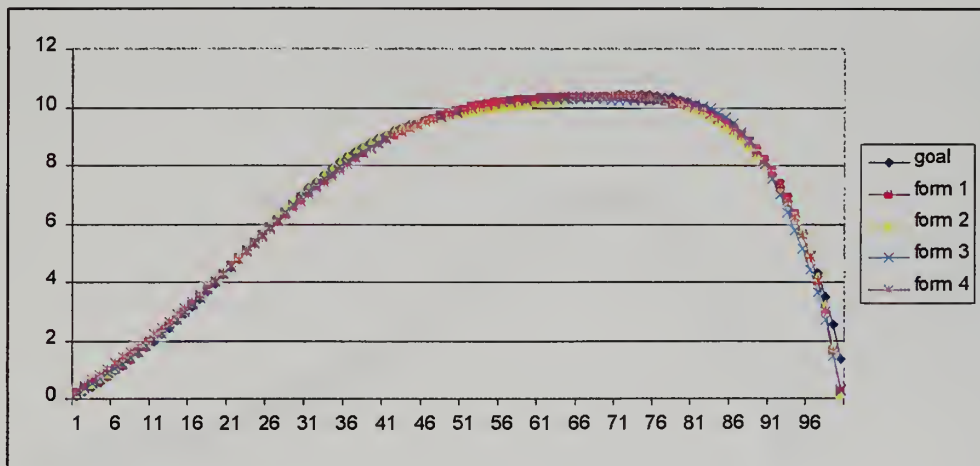The flat line indicates the minimum value of the best solution obtained.

Figure 20: Graphical Results for the AS-Test.
This figure shows results obtained for the AS-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 and
form 2 are the information curves for each form.

EI - Test (Electronic Information):

General Requirements:

    forms: 2;

    items: 20 each; and

    taxonomies: 4 (10,4,2,4 items in taxonomy 1 to 4)

Execution Specifics:

    repetitions: 100; and
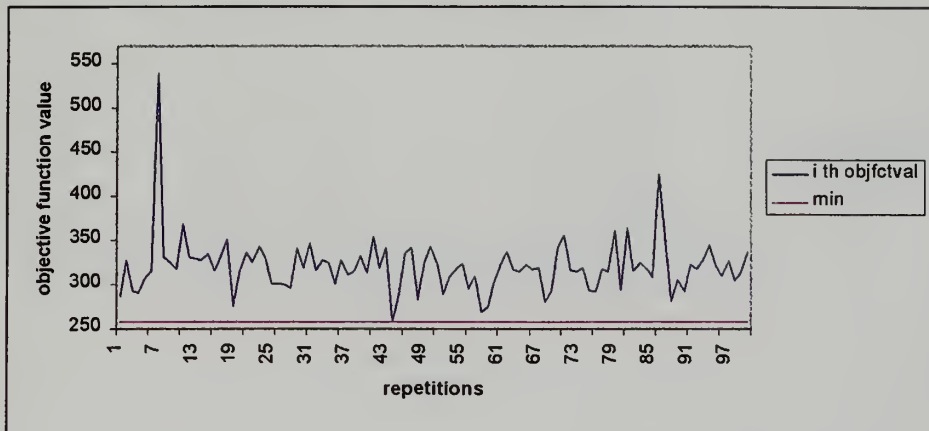
    objective function value: 227.10.

Figure 21: Objective Function Values for each Random Restart.
The flat line indicates the minimum value of the best solution obtained.
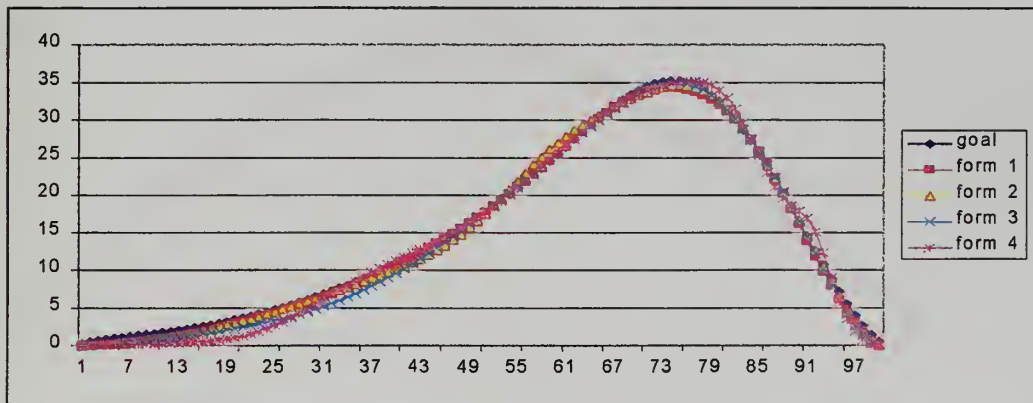

Figure 22: Graphical Results for the EI-Test.
This figure shows results obtained for the EI-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 and
form 2 are the information curves for each form.

GS - Test (General Science):

General Requirements:

    forms: 2;

    items: 25 each; and

    taxonomies: 12 (3,3,4,2,2,3,1,2,2,1,1,1).

Execution Specifics:

    repetitions: 100; and

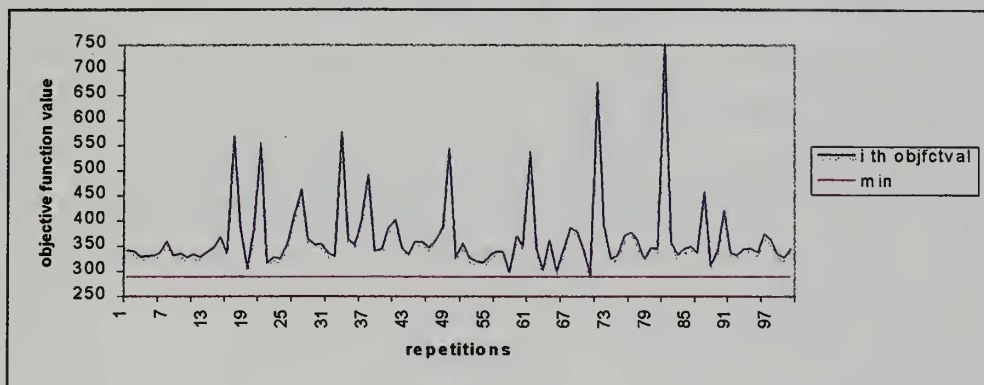   objective function value: 117.18.

Figure 23: Objective Function Values for each Random Restart.
The flat line indicates the minimum value of the best solution obtained.

Figure 24: Graphical Results for the GS-Test.
This figure shows results obtained for the GS-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 and
form 2 are the information curves for each form.

MC - Test (Mechanical Comprehension):

General Requirements:

forms: 4;

items: 25 each; and

taxonomies: 6 (11,2,2,2,4,4 items in taxonomy 1 to 6).

Execution Specifics:

repetitions: 100; and

objective function value: 47.8.

Figure 25: Objective Function Values for each Random Restart.
The flat line indicates the minimum value of the best solution obtained.

Figure 26: Graphical Results for the MC-Test.
This figure shows results obtained for the MC-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 to
form 4 are the information curves for each form.

MK - Test (Mathematical Knowledge):

General Requirements:

    forms: 4;

    items: 25 each; and

    taxonomies: 5 (3,5,9,7,1 items in taxonomy 1 to 5).

Execution Specifics:

    repetitions: 100; and

    objective function value: 257.94.

Figure 27: Objective Function Values for each Random Restart.
The flat line indicates the minimum value of the best solution obtained.

Figure 28: Graphical Results for the MK-Test.
This figure shows results obtained for the MK-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 to
form 4 are the information curves for each form.

WK - Test (Word Knowledge):

General Requirements:

    forms: 2;

    items: 35 each; and

    taxonomies: 2 (13,22 items in taxonomy 1 and 2).

Execution Specifics:

    repetitions: 100; and

    objective function value: 280.68.

Figure 29: Objective Function Values for each Random Restart.
The flat line indicates the minimum value of the best solution obtained.
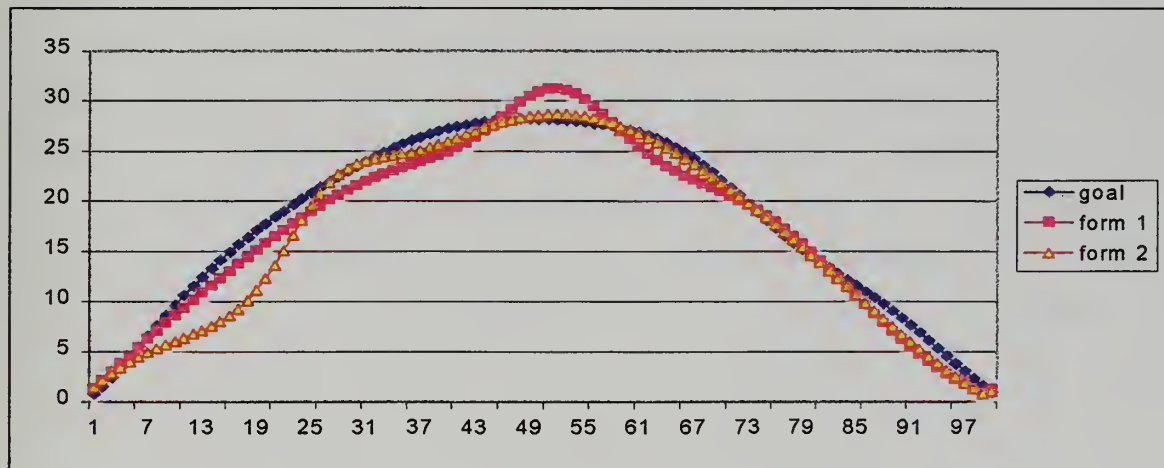
Figure 30: Graphical Results for the WK-Test.
This figure shows results obtained for the WK-Test with information on
the vertical axis and the percentiles on the horizontal axis. Form 1 and
form 2 are the information curves for each form.

D. DISCUSSION OF THE RESULTS

The optimization approach yields good results for the form assembly. The assembled forms for five out of the seven tests have information curves (form curves) that are very close to the goal curve and parallel to each other. In the EI- and WK-Test the form curves do not reach the goal curve in the lower half of the percentile range. Improving these forms by changing the weight of the parallel subgoal for the EI- and WK-Test to zero does not improve the shape of the form curves. Increasing the weight for the subgoal yields marginally more parallel forms, but increases the overall distance to the goal curve much more. Changing the bounds for the deviation variables has little effect. Discussions with DMDC indicate the item pools for the EI- and WK-Test are known to be "weak" since in their opinion, too many items were extracted for Computer Adaptive Testing. (See Wainer [1990] for a description of this relatively new method of testing.) They are working to restock these item pools.

The heuristic yields good results for the AR-, AS- and MC-Test. Results for GS-Test are not very parallel in the higher percentile range and results for the MK-Test are not very parallel in the lower percentile range. The form curves of the EI- and WK-Tests indicate the same deficiency in the item pool in the lower half of the percentile range as mentioned above. The number of repetitions has been increased to 1,000 in the AS- and EI-Test in order to see, whether the heuristic results can be improved. The objective function value decreased from 230 to 225 in the AS-Test and only from 227 to 226 in the EI-Test.

V. USING BOTH OPTIMIZATION AND HEURISTIC APPROACHES

A. USING THE HEURISTIC SOLUTION AS A BOUND

Table 4 summarizes a direct comparison of the objective function values where the heuristic solutions are converted to the objective function of the optimization model.

Test	Optimization Objective function value	Heuristic Objective function value
AR	932.33	1,840.57
AS	2,862.73	4,938.76
EI	9,561.94	10,676.62
GS	8,433.03	11,830.96
MC	1,187.11	3,377.86
MK	7,278.24	72,095.93
WK	5,188.42	17,883.56

Table 4: Comparison of the Results.
This table provides the objective function values for both the best heuristic solution and the best solution obtained solving the optimization model using the optimization model's objective function.

The optimization approach yields smaller objective function values than the heuristic as would be expected when using the optimization model's objective function as an evaluation. However, it is surprising that the differences are so great when the graphical results look similar. For the AR-Test, the heuristic approach in the percentile range 20 to 50 is not as parallel as in the optimization solution and this difference is responsible for nearly doubling the

objective function value. This is similar in the AS-Test, where form 2 is constantly below form 1. For the MK- and WK-Tests the corresponding objective function value is 9.9 and 3.5 times higher than the optimal result. The heuristic's solution having a higher objective function value for the MK-Test (Figure 15 and Figure 28) is caused by the parallel gap between form one and the three other forms in the lower percentiles combined with a high weight for the parallel subgoal. In the WK-Test the alternating behavior of the forms around each other in Figure 16 is similar to the heuristic solution (Figure 30). However, there is an obvious dominance of form one to form two in the lower percentile range. The heuristic solution for the MC-Test has a higher value than that of the optimization solution, however, the graphical result of the heuristic looks much better than the optimization. This is most likely due to the cancellation effect of positive and negative distances in Figure 14.

Using the heuristic solution as an upper bound for the objective function value when solving it using GAMS and OSL yields better results in almost all cases as shown in Table 5. Table 5 shows the MC-Test is an exception since the best solution with the heuristic bound is worse than without it. While this may happen due to OSL's branching choice within its branch and bound enumeration, having a bound should help in almost all cases.

Test	Objfctvalue (unbounded)	Objfctvalue (bounded)	Change of integrality gap (%)	Change of runtime (sec)
AR	932.33	905.13	- 3.1	- 7,690
AS	2,862.73	2,809.90	- 1.9	+ 642
EI	9,561.94	9,542.28	- 0.4	+ 110
GS	8,433.03	8,443.81	0.0	+ 337
MC	1,187.11	2,605.13	+1130.0	0
MK	7,278.24	6,532.59	-30.0	-47,567
WK	5,188.42	5,127.66	- 1.6	- 1,730

Table 5: Results of the Optimization Starting with the Best Heuristic
          Solution.

This table shows a comparison of the results for the optimization ap-
proach, when the heuristic solution bounds the objective function. A
negative number indicates an improvement in time or in the integrality
gap.

B.   CONCLUSIONS

This thesis demonstrates how using a linear mixed integer goal program can support DMDC's form assembly process. The developed heuristic is a good supplement that can be used with the optimization approach described. In some cases the heuristic solution yields good upper bounds for the optimization that can decrease the computation time.

C.   RECOMMENDATIONS

The optimization model should be extended to capture the other three ASVAB-Tests.

This heuristic algorithm should be considered a prototype. Experiments should be conducted with the objective function to find the most useful expression. While changing the objective function to match that currently implemented in the optimization model would be a natural first step, experimentation should be more expansive. The heuristic can easily accomodate a nonlinear objective function (an option not available in integer linear programming).

Further research can also be conducted to implement a heuristic for Computer Adaptive Testing.

LIST OF REFERENCES

Baker, H., Wall, J. 1996, *Interest Finder Form Assembly: The Influence of Pattern Matching to the Self-directed Search*, Technical Report 96-001, Defense Manpower Data Center, Seaside, CA.

Berger, F., Gupta, W., Berger, R. 1988, *Air Force Officer Qualifying Test (AFOQT) Form P: Test Construction*, Technical Report 88-30, Psycometrics Inc., Sherman Oaks, CA.

Brooke, A.; Kendrick, D.; Meeraus, A. 1992, *GAMS: A User's Guide, Release 2.25*, The Scientific Press, South San Francisco, CA.

Charnes, A. and Cooper, W.W. 1961, *Management Models and Industrial Applications of Linear Programming*, John Wiley & Sons, New York, NY.

Defense Manpower Data Center,DMDC, 1992, *Counselor Manual for the Armed Services Vocational Aptitude Battery, Forms 18/19* DoD Report 1304.12-L-ASTP-CM.

GAMS Development Corporation 1995, *GAMS - Installation and System Notes*, Washington, DC.

Lewis, P.A.W., Goodman, A.S. and Miller, J.M., 1969, *A Pseudo-Random Number Generator for the System/360*, IBM Sys. J., vol. no. 2: 136-146.

Lord, F.M., 1980, *Applications of Item Response Theory to Practical Testing Problems*, Lawrence Erlbaum Asociates, Hillsdale, NJ.

Lord, F.M., and Novick, M.R, 1968, *Statistical Theories of Mental Test Scores*, Addison-Wesley Publishing Company, Reading, MA.

Maier, M.H., and Truss, A.R., 1985, *Validity of the Armed Services Vocational Aptitude Battery Forms 8,9, and 10 with Applications to Forms 11, 12, 13, and 14*, Technical Report CNR-102, Center for Naval Analyses, Alexandria, VA.

Papadimitriou, C.H., and Steiglitz, K., 1982, *Combinatorial Optimization - Algorithms and Complexity*, Prentice-Hall, Englewoods Cliff, NJ.

Silicon Valley Software C^3 1991, *SVS C^3 Systems Manual*, San Mateo, CA.

US Air Force Human Resources Laboratory, 1973, *An Example of Cost Evasion from Use of Selection Tests*, Air Force Systems Command, Brooks Air Force Base, TX.

Toquarn, J.L., Corpe, V.A, Dunnette, M.D, 1991, *Literature Review: Cognitive Abilities-Theory, History, and Validity*, ARI Research Note 91-28, United States Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA.

Wainer, H., 1990, *Computerized Adaptive Testing: A Primer*, Lawrence Erlbaum Asociates, Hillsdale, NJ.

INITIAL DISTRIBUTION LIST

45

8.   German Armed Forces Command USA/Canada 1
      Attn: S1 StOffz
      11150 Sunrise Valley Drive
      Reston, VA 20191

9.   Bundesministerium der Verteidigung 1
      P II 4
      Postfach 1328
      53003 Bonn
      Germany